```
entry:
                                                                                                         call void @llvm.dbg.value(metadata ptr %func args, metadata !84, metadata
                                                                                                         ... !DIExpression()), !dbg !92
                                                                                                         %call = tail call i32 @initialise arrays(ptr noundef nonnull @ func .s253)
                                                                                                         ... #8, !dbg !93
                                                                                                         %call1 = tail call i32 @gettimeofday(ptr noundef %func args, ptr noundef
                                                                                                         ... null) #8, !dbg !94
                                                                                                         tail call void asm sideeffect ".inst 0x2520e020", ""() #8, !dbg !95, !srcloc
                                                                                                         tail call void asm sideeffect "dmb sy\0A\09orr x3,x3,x3\0A", "\sim{memory}"()
                                                                                                         ... #8, !dbg !98, !srcloc !100
                                                                                                         call void @llvm.dbg.value(metadata i32 0, metadata !86, metadata
                                                                                                         ... !DIExpression()), !dbg !101
                                                                                                         br label %for.cond2.preheader, !dbg !102
                                                                                                          for.cond2.preheader:
                                                                                                           %nl.038 = phi i32 [ 0, %entry ], [ %inc21, %for.cond.cleanup4 ]
                                                                                                          call void @llvm.dbg.value(metadata i32 %nl.038, metadata !86, metadata
                                                                                                          ...!DIExpression()),!dbg!101
                                                                                                          call void @llvm.dbg.value(metadata i32 0, metadata !88, metadata
                                                                                                          ...!DIExpression()),!dbg!103
                                                                                                           %0 = call i64 @llvm.vscale.i64(), !dbg !104
                                                                                                           %1 = mul i64 %0, 2, !dbg !104
                                                                                                           %2 = icmp uge i64 8192, %1, !dbg !104
                                                                                                          br i1 %2, label %pre.alc, label %Preheader.for.remaining.iterations, !dbg
                                                                                                           ... !104
                                                                                                                           Τ
                                                                                                                                                                  F
                                                        pre.alc:
                                                        \%8 = \text{call i64} \otimes \text{llvm.vscale.i64}
                                                        %9 = mul i64 %8, 2
                                                        %uniform.vector = call <vscale x 2 x i64>
                                                        ... @llvm.experimental.stepvector.nxv2i64()
                                                        %10 = urem i64 8192, %9
                                                        %total.iterations.to.be.vectorized = sub i64 8192, %10
                                                        %11 = insertelement <vscale x 2 x i64> poison, i64 %9, i64 0
                                                        %stepVector.update.values = shufflevector <vscale x 2 x i64> %11, <vscale x
                                                        ... 2 x i64> poison, <vscale x 2 x i32> zeroinitializer
                                                        %remaining.vector = add <vscale x 2 x i64> %uniform.vector,
                                                        ... %stepVector.update.values
                                                        %12 = getelementptr inbounds [8192 x i32], ptr @a, i64 0, i64 0, !dbg !122
                                                        %13 = getelementptr inbounds [8192 x i32], ptr @b, i64 0, i64 0, !dbg !130
                                                        %14 = call i64 @llvm.vscale.i64(), !dbg !122
                                                        %15 = mul i64 %14, 2, !dbg !122
                                                        %16 = getelementptr inbounds [8192 x i32], ptr @a, i64 0, i64 %15, !dbg !122
                                                        %17 = getelementptr inbounds [8192 x i32], ptr @b, i64 0, i64 %15, !dbg !130
                                                        %18 = load < vscale x 2 x i32>, ptr %12, align 8
                                                        %19 = load < vscale x 2 x i32 >, ptr %13, align 8
                                                        %21 = load < vscale x 2 x i32>, ptr %16, align 8
                                                        %22 = load < vscale x 2 x i32 >, ptr %17, align 8
                                                        %23 = icmp sqt < vscale x 2 x i32 > %21, %22
                                                        br label %alc.header
                                                          alc.header:
                                                          %24 = phi i64 [ 0, %pre.alc ], [ %45, %new.latch ]
                                                           %25 = phi <vscale x 2 x i64> [ %uniform.vector, %pre.alc ], [ %46,
                                                          ... %new.latch 1
                                                           \%26 = \text{phi} < \text{vscale x 2 x i64} > [\%\text{remaining.vector}, \%\text{pre.alc}], [\%47],
                                                           ... %new.latch ]
                                                           br i1 true, label %lane.gather, label %linearized
                                                                                                               F
lane.gather:
%27 = \text{call} < \text{vscale x 2 x i1} > \text{@llvm.aarch64.sve.ptrue.nxv2i1(i32 2)}
%28 = call <vscale x 2 x i64> @llvm.aarch64.sve.compact.nxv2i64(<vscale x 2
... x i1> %20, <vscale x 2 x i64> %uniform.vector)
%29 = call <vscale x 2 x i64> @llvm.aarch64.sve.compact.nxv2i64(<vscale x 2
... \times i1> %23, <vscale \times 2 \times i64> %remaining.vector)
%30 = xor <vscale x 2 x i1> %20, shufflevector (<vscale x 2 x i1>
... insertelement (<vscale x 2 x i1> poison, i1 true, i32 0), <vscale x 2 x i1>
... poison, <vscale x 2 x i32> zeroinitializer)
\%31 = xor < vscale x 2 x i1 > \%23, shufflevector (< vscale x 2 x i1 > \%23)
... insertelement (<vscale x 2 x i1> poison, i1 true, i32 0), <vscale x 2 x i1>
... poison, <vscale x 2 x i32> zeroinitializer)
%32 = call <vscale x 2 x i64> @llvm.aarch64.sve.compact.nxv2i64(<vscale x 2
... x i1> %30, <vscale x 2 x i64> %uniform.vector)
%33 = call <vscale x 2 x i64> @llvm.aarch64.sve.compact.nxv2i64(<vscale x 2
... x i1 > \%31, <vscale x 2 x i64 > \%remaining.vector)
%34 = call i64 @llvm.aarch64.sve.cntp.nxv2i1(<vscale x 2 x i1> %27, <vscale
... x 2 x i1 > \%20
%35 = call <vscale x 2 x i1> @llvm.aarch64.sve.whilelt.nxv2i1.i64(i64 0, i64
 .. %34)
%36 = call <vscale x 2 x i64> @llvm.aarch64.sve.splice.nxv2i64(<vscale x 2 x
... i1> %35, <vscale x 2 x i64> %28, <vscale x 2 x i64> %29)
%37 = call i64 @llvm.aarch64.sve.cntp.nxv2i1(<vscale x 2 x i1> %27, <vscale
... x 2 x i1 > \%23
%38 = call <vscale x 2 x i1> @llvm.aarch64.sve.whilelt.nxv2i1.i64(i64 0, i64
... %37)
%39 = call <vscale x 2 x i64> @llvm.aarch64.sve.splice.nxv2i64(<vscale x 2 x
... i1> %38, <vscale x 2 x i64> %29, <vscale x 2 x i64> %33)
%40 = call i64 @llvm.aarch64.sve.cntp.nxv2i1(<vscale x 2 x i1> %27, <vscale
... x 2 x i1 > \%30
%41 = call <vscale x 2 x i1> @llvm.aarch64.sve.whilelt.nxv2i1.i64(i64 0, i64
... %40)
%42 = call <vscale x 2 x i64> @llvm.aarch64.sve.sel.nxv2i64(<vscale x 2 x
... i1> %41, <vscale x 2 x i64> %32, <vscale x 2 x i64> %39)
%43 = xor < vscale x 2 x i1 > %41, shufflevector (< vscale x 2 x i1 > %41)
... insertelement (<vscale x 2 x i1> poison, i1 true, i32 0), <vscale x 2 x i1>
... poison, <vscale x 2 x i32> zeroinitializer)
%44 = \text{and} < \text{vscale x 2 x i1} > %43, %38
br label %alc.applied
                                                                                          alc.applied:
                                                                                                                   linearized:
                                                                                          br label %new.latch
                                                                                                                   br label %new.latch
                                                                     new.latch:
                                                                     %45 = add i64 %9, %24
                                                                     %46 = add <vscale x 2 x i64> %25, %stepVector.update.values
                                                                     %47 = add <vscale x 2 x i64> %26, %stepVector.update.values
                                                                     br i1 true, label %alc.header, label %middel.block
                                                                                        middel.block:
                                                                                         %condition = icmp eq i64 %10, 0
                                                                                         br i1 %condition, label %for.cond.cleanup4, label
                                                                                         ... %Preheader.for.remaining.iterations
                                                                                             Preheader.for.remaining.iterations:
                                                                                              %7 = phi i64 [ 0, %for.cond2.preheader ], [ %45, %middel.block ]
                                                                                              br label %for.body5
                                                                                     for.body5:
                                                                                      %indvars.iv = phi i64 [ %indvars.iv.next, %for.inc ], [ %7,
                                                                                     ... %Preheader.for.remaining.iterations ]
                                                                                      call void @llvm.dbg.value(metadata i64 %indvars.iv, metadata !88, metadata
                                                                                      ...!DIExpression()).!dbg!103
                                                                                      %arrayidx = getelementptr inbounds [8192 x i32], ptr @a, i64 0, i64
                                                                                      ... %indvars.iv, !dbg !122
                                                                                      %3 = load i32, ptr %arrayidx, align 4, !dbg !122, !tbaa !126
                                                                                      %arrayidx7 = getelementptr inbounds [8192 x i32], ptr @b, i64 0, i64
                                                                                      ... %inďvars.iv, ľdbg !130
                                                                                      %4 = load i32, ptr %arrayidx7, align 4, !dbg !130, !tbaa !126
                                                                                      %cmp8 = icmp sgt i32 %3, %4, !dbg !131
                                                                                      br i1 %cmp8, label %if.then, label %for.inc, !dbg !132
                                                                                                                                               F
                                                                   if.then:
                                                                    %arrayidx14 = getelementptr inbounds [8192 x i32], ptr @d, i64 0, i64
                                                                    ... %inďvars.iv, !ďbg !133
                                                                    %5 = load i32, ptr %arrayidx14, align 4, !dbg !133, !tbaa !126
                                                                    %mul = mul nsw i32 %5, %4, !dbg !135
                                                                    %sub = sub nsw i32 %3, %mul, !dbg !136
                                                                    call void @llvm.dbg.value(metadata i32 %sub, metadata !85, metadata
                                                                    ...!DIExpression()),!dbq!92
                                                                    %arrayidx16 = getelementptr inbounds [8192 x i32], ptr @c, i64 0, i64
                                                                    ... %indvars.iv, !dbg !137
                                                                    %6 = load i32, ptr %arrayidx16, align 4, !dbg !138, !tbaa !126
                                                                    %add = add nsw i32 %sub, %6, !dbg !138
                                                                    store i32 %add, ptr %arrayidx16, align 4, !dbg !138, !tbaa !126
                                                                    store i32 %sub, ptr %arrayidx, align 4, !dbg !139, !tbaa !126
                                                                    br label %for.inc, !dbg !140
                                                                                       for.inc:
                                                                                       %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1, !dbg !141
                                                                                       call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !88,
                                                                                       ... metadata !DIExpression()), !dbg !103
                                                                                       %exitcond.not = icmp eq i64 %indvars.iv.next, 8192, !dbg !142
                                                                                       br i1 %exitcond.not, label %for.cond.cleanup4, label %for.body5, !dbg!104,
                                                                                       ...!llvm.loop!143
                                                                for.cond.cleanup4:
                                                                 %call19 = tail call i32 @dummy(ptr noundef nonnull @a, ptr noundef nonnull
                                                                 ... @b, ptr noundef nonnull @c, ptr noundef nonnull @d, ptr noundef nonnull @e,
                                                                 ... ptr noundef nonnull @aa, ptr noundef nonnull @bb, ptr noundef nonnull @cc,
                                                                 ... i32 noundef 0) #8, !dbg !115
                                                                 %inc21 = add nuw nsw i32 %nl.038, 1, !dbg !116
                                                                 call void @llvm.dbg.value(metadata i32 %inc21, metadata !86, metadata
                                                                 ...!DIExpression()), !dbg!101
                                                                 %exitcond40.not = icmp eq i32 %inc21, 10, !dbg !117
                                                                 br i1 %exitcond40.not, label %for.cond.cleanup, label %for.cond2.preheader,
                                                                 ...!dbg!102,!llvm.loop!118
                                                                                                                             F
                                             for.cond.cleanup:
                                              tail call void asm sideeffect "dmb sy\0A\09orr x4,x4,x4\0A", "~{memory}"()
                                              ... #8, !dbg !105, !srcloc !107
                                              tail call void asm sideeffect ".inst 0x2520e040", ""() #8, !dbg !108,
                                              ...!srcloc!110
                                              %t2 = getelementptr inbounds %struct.args t, ptr %func args, i64 0, i32 1,
                                              ... !dbg !111
                                              %call23 = tail call i32 @gettimeofday(ptr noundef nonnull %t2, ptr noundef
                                              ... null) #8, !dbg !112
                                              %call24 = tail call i32 @calc checksum(ptr noundef nonnull @ func .s253)
```

... #8. !dba !113

ret i32 %call24, !dbg !114